Durham University

Durham Research Online

Deposited in DRO:

20 July 2021

Version of attached file:

Accepted Version

Peer-review status of attached file:

Peer-reviewed

Citation for published item:

Sun, Zhongtian and Harit, Anoushka and Yu, Jialin and Cristea, Alexandra and Al Moubayed, Noura (2021) 'A Generative Bayesian Graph Attention Network for Semi-supervised Classification on Scarce Data.', IEEE International Joint Conference on Neural Network (IJCNN2021) Virtual, 18-22 Jul 2021.

Further information on publisher's website:

https://doi.org/10.1109/IJCNN52387.2021.9533981

Publisher's copyright statement:

© 2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Additional information:

Use policy

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a link is made to the metadata record in DRO
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Please consult the full DRO policy for further details.

Durham University Library, Stockton Road, Durham DH1 3LY, United Kingdom
Tel : +44 (0)191 334 3042 | Fax : +44 (0)191 334 2971
https://dro.dur.ac.uk

A Generative Bayesian Graph Attention Network for Semi-supervised Classification on Scarce Data

1st Zhongtian Sun
Department of Computer Science
Durham University
Durham, UK
zhongtian.sun@durham.ac.uk

2nd Anoushka Harit
Department of Computer Science
Durham University
Durham, UK
anoushka.harit@durham.ac.uk

3rd Jialin Yu
Department of Computer Science
Durham University
Durham, UK
jialin.yu@durham.ac.uk

4th Alexandra I. Cristea
Department of Computer Science
Durham University
Durham, UK
alexandra.i.cristea@durham.ac.uk

5th Noura Al Moubayed
Department of Computer Science
Durham University
Durham, UK
noura.al-moubayed@durham.ac.uk

***Abstract*—This research focuses on semi-supervised classification tasks, specifically for graph-structured data under data-scarce situations. It is known that the performance of conventional supervised graph convolutional models is mediocre at classification tasks, when only a small fraction of the labeled nodes are given. Additionally, most existing graph neural network models often ignore the noise in graph generation and consider all the relations between objects as genuine ground-truth. Hence, the missing edges may not be considered, while other spurious edges are included. Addressing those challenges, we propose a Bayesian Graph Attention model which utilizes a generative model to randomly generate the observed graph. The method infers the joint posterior distribution of node labels and graph structure, by combining the Mixed-Membership Stochastic Block Model with the Graph Attention Model. We adopt a variety of approximation methods to estimate the Bayesian posterior distribution of the missing labels. The proposed method is comprehensively evaluated on three graph-based deep learning benchmark data sets. The experimental results demonstrate a competitive performance of our proposed model BGAT against the current state of the art models when there are few labels available (the highest improvement is 5%), for semi-supervised node classification tasks.**

***Index Terms*—semi-supervised learning, graph neural network, Bayesian neural network**

I. Introduction

Graph representation learning has recently drawn the attention of researchers from across various domains, including computer vision, natural language processing, knowledge graphs, and social networks [1]. Graphs are defined as $G = (V, E)$ where V is a set of nodes and E is a set of edges. As graphs can be irregular, arbitrary, non-Euclidean in structure and contain a rich range of values, they are good candidates to represent the complex knowledge (entities and relationships) existent in the real world [2]. Extensive graph neural network (GNN) based models, including graph convolution networks (GCNs) [3] and graph attention networks (GATs) [4] have been developed for unsupervised, semi-supervised and supervised learning cases. In the supervised setting, the training data is given with all class labels [5], and models pass the information from neighboring nodes and edges, to classify the central node, by optimizing a predefined loss function. Despite the learning abilities these models proved, their inference performance is compromised for semi-supervised tasks, when only limited labeled data is available [2]. Additionally, most existing studies [3], [6], [7] process input graphs as the ground truth, but neglect the fact that noise or spurious edges generated from model assumptions may be included, and thus lack robustness.

Addressing the above problems, [2] proposed a generative graph model to infer the joint posterior distribution of weights of a Graph Convolutional Network (GCN) and the graph structure of input. However, their posterior inference of the graph is mainly conditioned on the structure of the observed graph, and neglects the information from the node features. As the data may be correlated with the actual graph structure, the information of features is missing which resulted in a moderate performance [8], particularly under data-scarce situations.

In this paper, we introduce a generative model that considers the neighboring nodes' features and labels using an attention mechanism. We propose a Bayesian graph attention network (BGAT) model, to simultaneously boost the model's performance and robustness when solving semi-supervised classification.

In summary, the main contributions of this work are:

1) We propose a novel BGAT model combining GAT and the Bayesian method, which allows, to the best of our knowledge *accounting for uncertain information*, such as spurious and missing edges between nodes in a graph, by viewing the observed graphs, as generated from a parametric random graph family.
2) We demonstrate our model's improved performance at classification tasks under data-scarce (under-labeled data) situations.

II. Related Work

The Graph Neural Network (GNN) was initially introduced by [9] as updating nodes' states iteratively, until reaching stable states, by propagating discrete features from neighboring nodes. [10] elaborated it further, by extending recurrent models to deal with graphs, including directed, cyclic, or mixed graphs based on information diffusion and relaxation methods [11]. Later, [12] developed a new framework for GNNs, which contained a message-passing phase and a readout phase. It improved the performance at learning hierarchical graphical representations, by using sub-graphs. GNNs are effective at learning node representations, via feature propagation, but generally struggle to model the dependencies between various node labels. These methods usually assume a central node classification is conditionally independent of the features, but seldomly [8] model the joint distribution of the node labels and graph to extract more relationships among nodes.

Most existing studies take the input graph as a fixed observation and assume it represents the ground-truth information. In such cases, neural networks do not consider the uncertainty of information, including spurious edges and missing edges in graphs. A few studies [2], [8], [13] have understood the problem and proposed to solve it by using generative models. [13] introduced a Gaussian process-based approach in semi-supervised learning and [2] incorporated the stochastic block model and used Monte Carlo dropout [14]to represent model uncertainty in deep learning. Recently, [8] introduced a graph-based generative framework for semi-supervised learning, but they did not consider the data-scarce situations in active learning, which is the problem of interest in this paper.

Our study builds on the work by [2], as their idea of considering the uncertain graph information based on GCN is practical and effective for semi-supervised learning, particularly under data-scarce situations. However, they did not consider the specific weights of highly correlated features and the interaction between those features and graph structure as addressed above. By learning the influence among nodes, the attention mechanism could mitigate the problem of including spurious edges as addressed by [4]. We are interested in combining Bayesian methods with GAT to learn the joint distribution of labels, features and graph under data-scarce situations for semi-supervised node classification tasks.

III. Model Building - Preliminaries

A. Graph Attention Mechanism

The standard deterministic soft attention modules have been widely used in various neural networks. According to [15], the basic idea is to map a number of key-value pairs to the output. By indexing keys K and queries Q, attention modules obtain non-negative deterministic attention weights W through a softmax function, and the output values could then be aggregated in the attention computation as follows:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

where V are the values and d_k is the dimension of the keys. The softmax function is applied on the attention scores to calculate the attention distribution. The Graph Attention Network (GAT) was proposed by [4] and its power consisted in it considering the relative influence between neighboring nodes and central nodes, instead of the fixed weights used in GCN, based on attention mechanisms. It can be outlined as:

$$h_i^l = \sigma(\sum \alpha_{ij}^l W^t h_j^{l-1}) \tag{2}$$

Where α_{ij}^t is the attention score between node i and j and can be calculated as:

$$\alpha_{ij}^t = softmax(\sigma(a[Wh_i^{l-1}||Wh_j^{l-1}])) \tag{3}$$

σ is the LeakyReLU activation, Wh are the weights of node i and j in layer $l-1$. Operation $||$ is a concatenation and the softmax function used to sum up all neighbors of node i. Instead of performing single attention as above, employing multiple attention layers allows to jointly attend to information at different space [15] and has been used to solve node classification task as a graph aggregator by assuming each attention head's importance is the same [4].

More recently, [16] proposed a Gated Attention Network (GaAN) model, to further take into account the importance of each attention head, separately. The model performed well for both inductive node classification and traffic speed forecasting tasks, and the authors argued that it could also be possibly extended to integrate edge features for massive graphs.

B. Bayesian Neural Network

According to Seedat and Kanan [17], Bayesian neural networks represent uncertainty by formulating a network's parameters in the manner of a probabilistic distribution. The weight matrix can be modelled as random variables that $p(W_l)$ in layer l can be defined as $W_l \sim N(0, \mathrm{I})$, by introducing the standard matrix Gaussian prior distribution with bias vector b_l. The classification task can be formulated as:

$$p(y|x, W_l) = categorical(exp(\widehat{f})/\sum_{d'} exp(\widehat{f}_{d'})) \tag{4}$$

where $\widehat{f} = \widehat{f}(x, W_l)$ is a random output, given input x and a random weight of a neural network W_l. Then a softmax function is added to obtain a multinomial probability distribution. Considering the posterior distribution of the weight matrix W, the predictive function for a new point x* can be formulated as:

$$p(y|x^*, X, Y) = \int p\left(y|x^*, w\right) p(w|X, Y) dw \tag{5}$$

As the functional form of a neural network is difficult to integrate, the exact calculation of the model posterior $p(w|X, Y)$ is generally intractable [18] and cannot usually be analysed in a close form. Therefore, [19] introduced an approximating variational distribution $q(w)$ and minimised the Kullback–Leibler (KL) divergence to approximate the predictive distribution:

$$p(y|x^*, X, Y) = \int p\left(y|x^*, w\right) q(w) dw \tag{6}$$

The variational inference of equation (6) can be further approximated using the Monte Carlo dropout method [14], which accounts for model uncertainty in deep learning. We can draw samples from the approximate posterior and average the weight matrix W of the network with T stochastic forward passes:

$$p(y|x^*, X, Y) \approx \frac{1}{T}\sum_{t=1}^{T} p(y|x^*, W_1^t, ..., W_l^t) \tag{7}$$

C. Mixed Membership Stochastic Block Model

The mixed membership stochastic block model (MMSBM) [20] is a popular framework for community detection [21], which considers a graph $G(V, E)$ and the associated adjacency matrix A. Assuming there are n nodes in the graph, denoted by $x_1, \ldots, x_n$. An adjacency matrix A for this graph is an n by n dimensional matrix. If there is no connection between node x_p and node x_q, $A(p, q) = 0$ otherwise $A(p, q) = 1$. The MMSBM models the adjacency matrix A in a Bayesian hierarchical framework. According to a very recent work of [22], the absence or presence of a link between any pair of nodes (x_p, x_q) is described by a Bernoulli distribution B with a latent group membership $z_{p,q,1}$ $z_{p,q,2}$:

$$A(p, q) \mid z_{p,q,1}, z_{p,q,2}, B \sim \text{Bernoulli}\left(z_{p,q,1}^T B z_{p,q,2}\right) \tag{8}$$

The Bernoulli probability matrix B has a K by K dimension, which, for community detection, represents the number of communities in the data. As we focus on undirected graphs, $z_{p,q,1}$ means the node x_p is interacting with x_q where both nodes are K-dimensional vectors, where only one element equals to one. It can be denoted as $z = [z_1, ...z_K]^T$, indicating the corresponding community the node belongs to (the rest being zero). According to [22], the joint distribution of latent group memberships of nodes Z and data X is:

$$\begin{aligned} p\left(X, Z_1, Z_2, \pi \mid \alpha, B\right) = & \prod_{p,q} p_1\left(X(p,q) \mid z_{p,q,1}, z_{p,q,2}, B\right) \\ & p_2\left(z_{p,q,1} \mid \pi_p\right) p_2\left(z_{p,q,2} \mid \pi_q\right) \prod_{p} p_3\left(\pi_p \mid \boldsymbol{\alpha}\right) \end{aligned} \tag{9}$$

p_1 is the Bernoulli distribution (β) which refers to the possibility of a link between two nodes. p_2, p_3 are prior of latent group membership and the prior of the former one, with multinomial and Dirichlet distributions, respectively.

IV. Bayesian Graph Attention Network

A. Learning Attention Using the Bayesian Framework

We consider a semi-supervised learning problem and model it with conditional probability using parametrization, including the graph's attention. The deterministic attention weights are transformed into a distribution, making it straightforward and requiring minimal changes to the standard attention model. We can adapt a pre-trained standard attention model for variational fine-tuning. For backpropagate through stochastic nodes, re-parametrization trick [23] can be used to construct the distribution. Learning attention distribution as a variational inference helps in constructing a re-parametrizable attention distribution. The graph structure is encoded in the attention masks, so that nodes can only attend to the neighborhood's features in the graph.

B. Methodology

As aforementioned, the motivation for introducing the Bayesian Graph Attention model derives from the Bayesian Graph Convolutional Neural Network (BGCN) model [2]. We use a building block layer to construct an arbitrary graph attention network (through stacking this layer) and apply a Bayesian approach. In this paper, the Bayesian approach views a graph as a realization from a parametric family of random graphs based on the known labels of nodes, Y and structures from observed graphs. The joint posterior of the weights in GAT, the parameters of the random graph and the remaining unknown node labels are the target inference. By marginalization, the graph parameters, posterior estimation of the labels could be inferred and obtained. Then we combine the posterior of labels and attention of nodes and implement a softmax function to obtain the final output. The posterior probability of labels is formulated as:

$$\begin{aligned} p\left(\mathbf{Z} \mid \mathcal{G}_{obs}, \mathbf{X}, \mathbf{Y}\right) = \int & p(\mathbf{Z} \mid W, \mathcal{G}_r, \mathbf{X}) p\left(W \mid \mathbf{Y}, \mathbf{X}, \mathcal{G}_r\right) \\ & p(\mathcal{G}_r \mid \zeta) p\left(\zeta \mid \mathcal{G}_{obs}\right) dW d\mathcal{G}_r d\zeta \end{aligned} \tag{10}$$

ζ is the parameter that describes a family of random graphs G_r, which can be derived from the observed graph G_{obs} using the MMSBM random graph model. W is the sampled weights of BGAT over the random graphs G_r by approximating variational inference via Monte Carlo dropout as aforementioned in III-B. $\mathbf{Y}$ is the known label of nodes and later GAT will take (X, G) as input to infer the unknown labels of nodes. A softmax function will be added to the output of GAT to model $p(Z|X, Y, G_{obs})$ in a categorical distribution. Figure 1 provides a detailed schematic of our Bayesian GAT model.

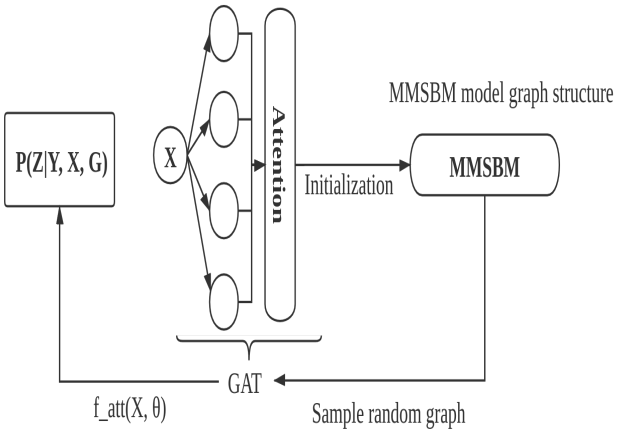

Fig. 1. Overview of BGAT

Since the highly non-linear nature of likelihood leads to intractable computation of posterior in the equation (10), we can use variational inference [14], [24], [25] or MCMC [26], [27] to approximate the posterior of $p(W|Y, X, G_r)$. According to [28], averaging the weights of the network is an approximate way of Monte Carlo dropout. The weight metrices W can then be sampled from $p\left(W \mid \mathbf{Y}, \mathbf{X}, \mathcal{G}_r\right)$ using Monte Carlo dropout given the sampled graphs generated from $p(\mathcal{G}_r|\zeta)$. To model $p(\zeta|G_{obs})$, parametric random graph generation models, such as degree corrected block model [29] and mixed membership stochastic block model [20] could be considered. In summary, the Monte Carlo approximation of equation (10) is:

$$p\left(\mathbf{Z} \mid \mathbf{Y}, \mathbf{X}, \mathcal{G}_{\text{obs}}\right) \approx \frac{1}{V} \sum_{v}^{V} \frac{1}{N_G S} \sum_{a=1}^{N_C} \sum_{s=1}^{S} p\left(\mathbf{Z} \mid W_{s,a}, \mathcal{G}_a, \mathbf{X}\right) \tag{11}$$

G_a is the graph sampled from the random graphs G_r and the weight matrix $W_{s,a}$ is sampled from $p\left(W \mid \mathbf{Y}, \mathbf{X}, \mathcal{G}_r\right)$ based on G_a. For the Bayesian GAT, we use a similar Mixed Membership Stochastic Block Model (MMSBM) setting used in Bayesian GCN [2] for the graph and learn its parameter $\zeta = \{\beta, z\}$ using stochastic optimization to maximize the posterior of β and z based on the observed graph G_{obs}.

As addressed in section III-C, the MMSBM model is used to model the random graph based on the observed graphs, which helps us establish a strong community structure between nodes and determine which community node may belong to. If any two nodes belong to the same community, meaning they have the same label, and it is highly likely to have a link between them, compared to when the two nodes belong to different communities [20]. β is to denote the possibility that there is a link between any two nodes and z is the parameter for the community membership probability distribution of nodes, and the priors of them are Beta and Dirichlet distribution, respectively.

The posterior probability of labels $p(Z|Y, X, G_{obs})$ is modeled as a K-dimensional categorical distribution, where K is the number of classes/communities of the data. In GAT, a weight matrix W containing the F dimensions of features of the nodes is introduced as an initial linear transformation, and then self-attention will be applied, to compute attention coefficients, as the relative influence of node j's features on node i, defined by [4]:

$$e_{ij} = a\left(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j\right) \tag{12}$$

The Bayesian inference of attention can be generalised to a stochastic generative process [30], with input x and the parameter of the posterior distribution of the attention θ:

$$\theta \sim p(\theta, G), \quad z = f_{att}\left(x; \theta\right) \tag{13}$$

Where z is the output of the attention model and the Bayesian inference for the attention and labels can be formulated as $p(z|x, G)$. Hence we model the posterior of labels and attention nodes, which allows us to incorporate the features of the nodes and trained labels in the graph inference process.

This study uses the Markov Chain Monte Carlo (MCMC) method to approximate the posterior of labels and model the posterior using a categorical distribution to establish the community membership among nodes. We then model the posterior of labels and learned attention of nodes by applying the softmax function to the output of GAT. The attention improves the model's expressiveness and the Bayesian GAT model leverages deterministic self-attention layers to process node features for graph node classification. The graph structure is encoded in the attention masks, so nodes can only attend to their neighborhoods' features in the graph.

V. Experiment

Experiments implemented to demonstrate the effectiveness of our model are described in this section. Specifically, we aim to answer the following research questions:

RQ1 How to improve graph representation learning models for semi-supervised classification tasks given scarce data?

RQ2 How to improve the robustness of the graph representation learning model to noise by considering uncertainty?

A. Datasets

As mentioned, we perform a semi-supervised node classification task on three citation datasets: Cora, CiteSeer, and Pubmed [31]. The details of each dataset is summarised in Table I. In these datasets, each node represents a scientific document and if anyone paper cites the other, there will be an undirected edge between them, shown in Figure 3 and 4, where Citeseer is more decentralized compared with Cora. We do not consider the direction of the citation here. Each node has a sparse feature vector (keywords of the document) and the label describes the topic of the document. For instance, each node has 1433 dimensions of features attached to it, represented as 0 or 1 in Cora and the node label in the last column represents the topic/community that document belongs to, as shown in

Figure 2. Please note that we only have access to few nodes per class during training to infer labels for other nodes.

TABLE I
DATASET SUMMARY

Datasets	Cora	Citeseer	Pubmed
Nodes	2708	3327	19717
Edges	5429	4732	44338
Communities	7	6	3
Features	1433	3703	500
Features Type	Binary	Binary	TF/IDF
Average Degree	4	2	3

ID	w_0	w_1	w_2	w_3	w_4	w_5	w_1430	w_1431	w_1432	subject
31336	0	0	0	0	0	0	0	0	0	Neural_Networks
1061127	0	0	0	0	0	0	0	0	0	Rule_Learning
1106406	0	0	0	0	0	0	0	0	0	Reinforcement_Learning
13195	0	0	0	0	0	0	0	0	0	Reinforcement_Learning
37879	0	0	0	0	0	0	0	0	0	Probabilistic_Methods
...	...	...	...	...	...	...	...	...	...	...
1128975	0	0	0	0	0	0	0	0	0	Genetic_Algorithms
1128977	0	0	0	0	0	0	0	0	0	Genetic_Algorithms
1128978	0	0	0	0	0	0	0	0	0	Genetic_Algorithms
117328	0	0	0	0	1	0	0	0	0	Case_Based
24043	0	0	0	0	0	0	0	0	0	Neural_Networks

Fig. 2. Layout of the Cora Dataset

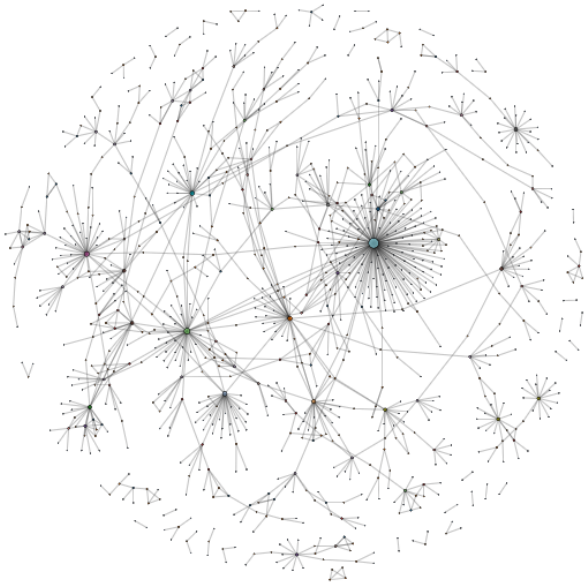

Fig. 3. Visualisation of the Cora Dataset

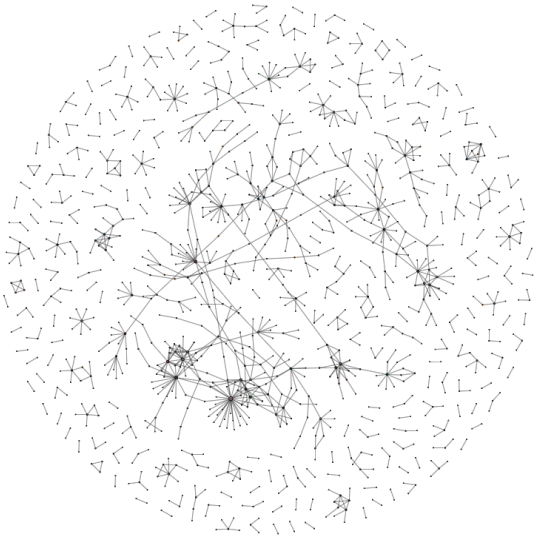

Fig. 4. Visualisation of the Citeseer Dataset

B. Baselines

We consider three widely applied graph learning models and one previous work on semi-supervised learning under data-scarce situations as baselines: ChebyNet, GCN, GAT and BGCN.

- **ChebyNet** A spectral CNN-based model uses Chebyshev polynomials to approximate and localize filters of graphs [32].
- **GCN** A spectral CNN-based method that deploys the first-order approximation of ChebNet and assigns a non-parametric weight of neighborhood to central nodes [3].
- **GAT** An attention mechanism-based neural network method that considers the weight of neighbor information to central nodes [4].
- **BGCN** A Bayesian GCNN framework that considers the randomness of input graphs by incorporating the Bayesian method with GCN [2].

C. Model Settings

The hyper-parameters of BGAT are borrowed from the experiments of GAT [4] and BGCN [2]. We use two layers and the number of hidden units is 16, with a 50 percent dropout rate at each layer. Learning rate is 0.01 and the L2 regularisation parameter is 0.0005. In addition, the hyper-parameters associated with the Mixed Membership Stochastic Block Model (MMSBM) inference are: n =500, ε_0 = 1, τ = 1024, κ = 0.5, η = 1, α = 1 and ρ= 0.001.

Three different experimental settings for a semi-supervised classification task have been considered, where 5, 10 and 20 labels per class are available in the training set, to infer labels for the others. The partitioning of the data into 20 labels per class is set the same as in [2], whereas in the other two cases, the training sets are constructed by considering the first 5 or 10 labels from the previous partition.

D. Experimental Results

TABLE II
THE AVERAGE PREDICTION ACCURACY OF MODELS IN CORA

Random split	5 labels	10 labels	20 labels
ChebyNet	61.7±6.8	72.5±3.4	78.8±1.6
GCN	70.0±3.7	76.0±2.2	79.8±1.8
GAT	70.4±3.7	76.6±2.8	79.9±1.8
BGCN	74.6±2.8	77.5±2.6	80.2±1.5
BGAT	$\mathbf{74.8 \pm 4.5}$	$\mathbf{78.8 \pm 2.8}$	$\mathbf{84.3 \pm 1.8}$

TABLE III
AVERAGE PREDICTION ACCURACY OF MODELS IN CITESEER

Random split	5 labels	10 labels	20 labels
ChebyNet	58.5±4.8	65.8±2.8	67.5±1.9
GCN	58.5±4.7	65.4±2.6	67.8±2.3
GAT	56.7±5.1	64.1±3.3	67.6±2.3
BGCN	63.0±4.8	69.9±2.3	71.1±1.8
BGAT	$\mathbf{68.6 \pm 4.6}$	$\mathbf{71.4 \pm 2.6}$	$\mathbf{74.2 \pm 1.6}$

TABLE IV
AVERAGE PREDICTION ACCURACY OF MODELS IN PUBMED

Random split	5 labels	10 labels	20 labels
ChebyNet	62.7±6.9	68.6±5.0	74.3±3.0
GCN	69.7±4.5	$\mathbf{73.9 \pm 3.4}$	$\mathbf{77.5 \pm 2.5}$
GAT	68.0±4.8	72.6±3.6	76.4±3.0
BGCN	70.2±4.5	73.3±3.1	76.0±2.6
BGAT	$\mathbf{71.4 \pm 4.7}$	72.3±3.4	74.5±2.4

The mean and standard deviation of the proposed method's test accuracy versus the baselines are shown above in Tables II, III and IV. The proposed model achieves a better performance in all bar 2 cases. For instance, the proposed model improves more than 4% and 5% of the test set accuracy in the Cora and Citeseer data sets, when there are 20 and 5 labels available, respectively. The results support the improvements introduced by our model for classification tasks under data-scarce situations. Nevertheless, the proposed model does not reach the highest accuracy in the Pubmed dataset, for the cases of 10 and 20 labels per community. GCN presents the most expressive power for the Pubmed dataset. One possible explanation is that there are more low-pass subgraphs in Pubmed and the repeated graph propagation is the primary source of the expressive power of GCN [33]. Another possible reason is that the MMSBM model may not be appropriate for data with a heavy-tailed degree distribution [2].

VI. CONCLUSION

This paper introduces a novel method for graph-based semi-supervised learning, which allows for considering *uncertainty in the graph generation process*. We present how to incorporate MMSBM with a graph attention mechanism and examine our model on three graph-based deep learning benchmark datasets. The results demonstrate that the proposed model outperforms other graph-based semi-supervised learning methods, when there are only a few labels of the nodes known for classification tasks in most settings. Given the robustness and performance of BGAT, it could be used as a new baseline in future generative graph learning.

VII. FUTURE WORK

There are several potential extensions to our work that could be addressed as future study. One is to investigate how to extend the generative models, by accounting for more graph structure information to other graph-based learning tasks. For instance, the direction of the citation could be considered and modelled in the graph generation process. Moreover, extending the method's expressive power with sub-structure counting could also be insightful, from the application perspective. Finally, extending the model with more scalable techniques would allow us to perform practical inference over large-scale graphs under data-scarce situations.

REFERENCES

[1] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: A survey," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[2] Y. Zhang, S. Pal, M. Coates, and D. Ustebay, "Bayesian graph convolutional neural networks for semi-supervised classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 5829–5836.

[3] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[4] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[5] B. Liu, "Statistical learning for networks with node features," Ph.D. dissertation, The University of Michigan, 2019.

[6] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *arXiv preprint arXiv:1706.02216*, 2017.

[7] H. Gao, Z. Wang, and S. Ji, "Large-scale learnable graph convolutional networks," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1416–1424.

[8] J. Ma, W. Tang, J. Zhu, and Q. Mei, "A flexible generative framework for graph-based semi-supervised learning," in *Advances in Neural Information Processing Systems*, 2019, pp. 3281–3290.

[9] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 2. IEEE, 2005, pp. 729–734.

[10] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2008.

[11] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[12] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," *arXiv preprint arXiv:1704.01212*, 2017.

[13] Y. C. Ng, N. Colombo, and R. Silva, "Bayesian semi-supervised learning with graph gaussian processes," in *Advances in Neural Information Processing Systems*, 2018, pp. 1683–1694.

[14] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, 2016, pp. 1050–1059.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[16] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, and D.-Y. Yeung, "Gaan: Gated attention networks for learning on large and spatiotemporal graphs," *arXiv preprint arXiv:1803.07294*, 2018.

[17] N. Seedat and C. Kanan, "Towards calibrated and scalable uncertainty representations for neural networks," *arXiv preprint arXiv:1911.00104*, 2019.

[18] R. van de Schoot, S. Depaoli, R. King, B. Kramer, K. Märtens, M. G. Tadesse, M. Vannucci, A. Gelman, D. Veen, J. Willemsen *et al.*, "Bayesian statistics and modelling," *Nature Reviews Methods Primers*, vol. 1, no. 1, pp. 1–26, 2021.

[19] Y. Gal and Z. Ghahramani, "Bayesian convolutional neural networks with bernoulli approximate variational inference," *arXiv preprint arXiv:1506.02158*, 2015.

[20] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, "Mixed membership stochastic blockmodels," *Journal of machine learning research*, vol. 9, no. Sep, pp. 1981–2014, 2008.

[21] Y. Zhang and A. Ramesh, "Struct-mmsb: Mixed membership stochastic blockmodels with interpretable structured priors," *arXiv preprint arXiv:2002.09523*, 2020.

[22] W. Huang, Y. Liu, Y. Chen *et al.*, "Mixed membership stochastic blockmodels for heterogeneous networks," *Bayesian Analysis*, vol. 15, no. 3, pp. 711–736, 2020.

[23] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick," *arXiv preprint arXiv:1506.02557*, 2015.

[24] S. Sun, C. Chen, and L. Carin, "Learning structured weight uncertainty in bayesian neural networks," in *Artificial Intelligence and Statistics*, 2017, pp. 1283–1292.

[25] C. Louizos and M. Welling, "Multiplicative normalizing flows for variational bayesian neural networks," *arXiv preprint arXiv:1703.01961*, 2017.

[26] R. M. Neal, *Probabilistic inference using Markov chain Monte Carlo methods*. Department of Computer Science, University of Toronto Toronto, Ontario, Canada, 1993.

[27] A. Korattikara Balan, V. Rathod, K. P. Murphy, and M. Welling, "Bayesian dark knowledge," *Advances in Neural Information Processing Systems*, vol. 28, pp. 3438–3446, 2015.

[28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[29] L. Peng, L. Carvalho *et al.*, "Bayesian degree-corrected stochastic blockmodels for community detection," *Electronic Journal of Statistics*, vol. 10, no. 2, pp. 2746–2779, 2016.

[30] B. An, J. Lyu, Z. Wang, C. Li, C. Hu, F. Tan, R. Zhang, Y. Hu, and C. Chen, "Repulsive attention: Rethinking multi-head attention as bayesian inference," *arXiv preprint arXiv:2009.09364*, 2020.

[31] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI magazine*, vol. 29, no. 3, pp. 93–93, 2008.

[32] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in neural information processing systems*, vol. 29, pp. 3844–3852, 2016.

[33] F. Wu, T. Zhang, A. H. d. Souza Jr, C. Fifty, T. Yu, and K. Q. Weinberger, "Simplifying graph convolutional networks," *arXiv preprint arXiv:1902.07153*, 2019.